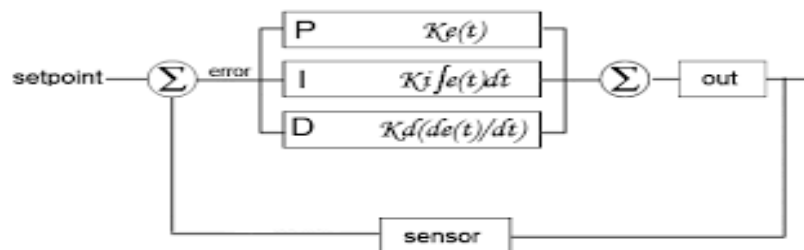


## BAB II

### DASAR TEORI

#### 2.1 *Proportional Integral Derivative* ( PID )

*Proportional Integral Derivative* ( PID ) kontroler merupakan kontroler untuk menentukan kepresisian suatu sistem instrumentasi dengan karakteristik adanya umpan balik / feedback pada sistem tersebut. Komponen PID terdiri dari 3 jenis, yaitu Proportional, Integral dan Derivatif. Ketiga dapat dipakai bersamaan maupun sendiri-sendiri, tergantung dari respon yang kita inginkan terhadap suatu plant.



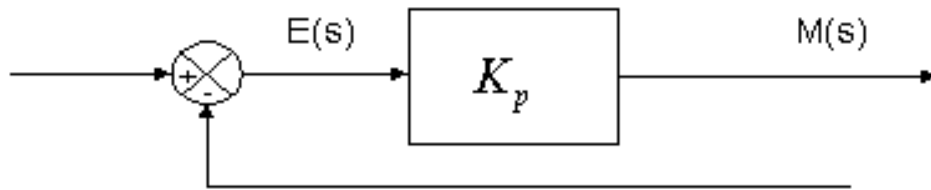
**Gambar 2.1** Blok Diagram PID

Sumber : <https://www.isa.org>

Ada 3 macam control PID yaitu kontroler PI, PD, dan PID. PI adalah kontrol yang menggunakan komponen proportional dan Integratif. PD adalah kontrol yang menggunakan komponen *Proportional* dan *Derivatif*. Dan PID adalah kontrol yang menggunakan komponen *Proportional*, *Integratif*, dan *Derivatif*.

##### 2.1.1 Kontroler Proporsional

Kontroler proporsional memiliki karakteristik bahwa keluarannya berupa variabel yang dikontrol berubah sebanding dengan masukan yang berupa variabel selisih antara masukan acuan dengan variabel termanipulasi atau keluarannya nyata dari plant.

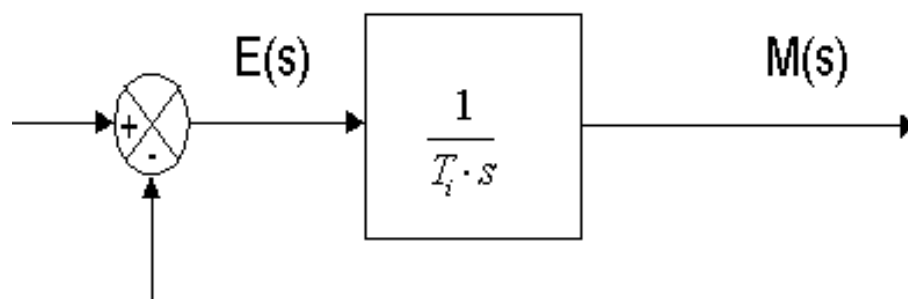


**Gambar 2.2 Kontrol Proporsional**

Sumber : <http://elektroindonesia.com>

### 2.1.2 Kontroler Integral

Laju perubahan nilai output dari kontroler integral sebanding dengan nilai masukan. Masukan sistem berupa variabel selisih antara masukan acuan (referensi) dengan variabel termanipulasi atau output nyata dari plant. Jadi, jika selisih acuan dengan output nyata besar maka perubahan nilai output juga besar, artinya aktuator akan mengejar selisih tersebut, sehingga diharapkan selisihnya semakin kecil. Karakteristik dan diagram blok kontroler integral dibandingkan gambar 2.3 diperlihatkan pada dengan kontroler proporsional, pemakaian kontroler integral relatif lebih baik dalam hal memperkecil selisih antara masukan acuan dengan output nyata. Dengan demikian, kontroler integral akan mendorong sistem yang dikontrol untuk mencapai output yang diinginkan, sehingga selisihnya semakin kecil.



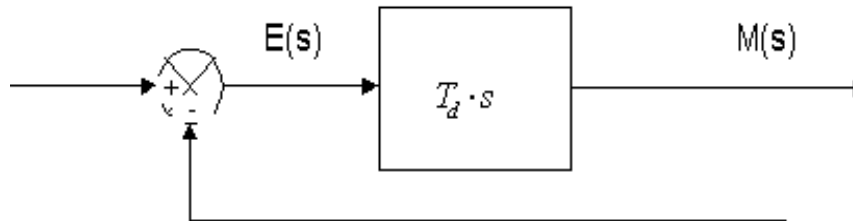
**Gambar 2.3 Kontroler Integral**

Sumber : <http://elektroindonesia.com>

### 2.1.3 Kontroler Derivatif

Penggunaan kontroler proporsional saja dalam sistem kontrol kadang-kadang menyebabkan respon sistem melebihi masukan acuannya. Misalnya dalam keadaan *overshoot*, untuk mengurangi pwm atau menghindari kondisi ini maka digunakan kontroler derivatif. masukan ke kontroler derivatif berupa perubahan selisih antara keluaran nyata dan masukan acuannya atau kecepatan kesalahan, sehingga apabila selisih antara keluaran nyata dan masukan acuannya semakin

besar maka kontroler mengirim sinyal ke aktuator yang semakin besar pula. Dengan demikian, nilai output yang melebihi nilai acuannya ditekan sekecil mungkin. Respon kontroler ini untuk masukan tangga dan masukan lereng.



**Gambar 2.4** *Kontroler Derivatif*

Sumber : <http://elektroindonesia.com>

## 2.2 *Laboratory Virtual Instrumentation Engineering Workbench (LabVIEW)*

*Laboratory Virtual Instrumentation Engineering Workbench* (LabVIEW) adalah suatu bahasa pemrograman berbasis grafis yang menggunakan ikon atau gambar sebagai ganti teks untuk menciptakan aplikasi. Berlawanan dengan bahasa pemrograman berbasis teks, di mana instruksi menentukan pelaksanaan program, LabVIEW menggunakan pemrograman dataflow, yang mana alur data menentukan eksekusi dari program. Tampilan pada LabVIEW menirukan instrumen secara virtual.

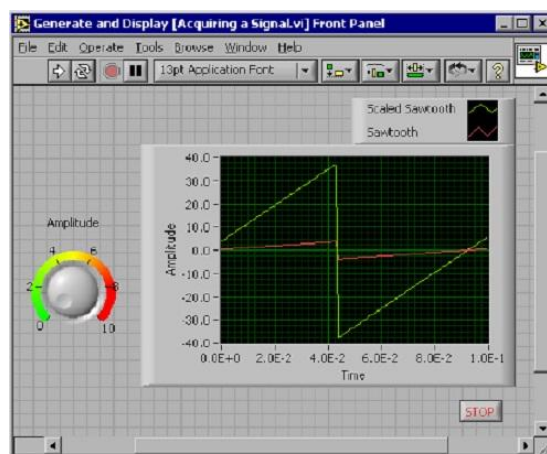
Program LabVIEW disebut sebagai virtual instruments atau VIs sebab operasi dan penampilannya meniru instrumen secara fisik, seperti multimeter dan osiloskop. LabVIEW berisi berbagai macam peralatan untuk menghasilkan ketelitian, tampilan, dan menyimpan data, seperti halnya perlengkapan untuk membantu melakukan pemecahan masalah pengkodean. Setiap Virtual Instrument (VI) menggunakan fungsi-fungsi yang menggerakkan masukan dari pemakai antarmuka atau sumber lain dan menampilkan informasi itu.

Di dalam LabVIEW juga di lengkapi dengan peralatan pendukung sistem kontrol dan pemodelan sistem seperti *Proportional Integral Derivative* ( PID ) maupun *Fuzzy Logic Controller* ( FLC ) yang memudahkan pengguna dalam merancang dan membuat sistem kontrol yang diinginkan.

### 2.2.1. Front Panel

Front Panel digunakan untuk berinteraksi dengan pengguna pada saat program berjalan. Pengguna dapat mengontrol program, mengubah masukan dan memantau data secara *real-time*. Kontrol digunakan untuk pengaturan masukan seperti mengubah nilai, menekan saklar atau menghentikan program. Kontrol digunakan untuk melakukan simulasi masukan data ke blok diagram. Indikator digunakan sebagai keluaran seperti, intensitas cahaya, data, suara, dan info lainnya. Indikator dapat berupa grafik, LED, dan tampilan dalam bentuk grafik. Indikator melakukan simulasi devais instrumen keluaran dan memperagakan data yang dihasilkan.

Front panel mengolah tampilan dari pemrograman LabVIEW agar mudah dimengerti oleh pengguna dan sebagai sarana interaksi dengan program yang berjalan. Objek yang berada pada front panel juga dapat dipindah dan diubah ukurannya sesuai kehendak pengguna.



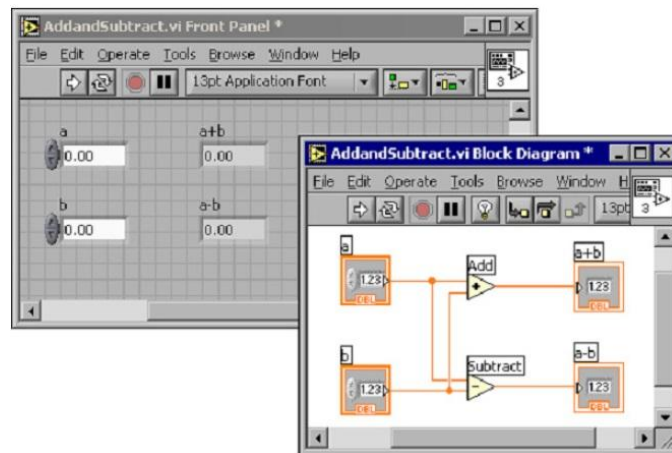
**Gambar 2.5** Tampilan *Front Panel* pada *LabVIEW*

Sumber : [Labview User Manual](#)

### 2.2.2. Block Diagram

Diagram blok berisi kode sumber yang bersifat grafis. Di dalam diagram blok, objek dari front panel berbentuk terminal yang dapat dihubungkan. Sehingga di dalam blok diagram terdapat berbagai komponen yang mewakili bagian front panel dan juga fungsi tertentu dalam perancangan. Blok diagram merupakan kode sumber yang menjadi inti dalam program VI, perancangan yang terdapat pada blok diagram berupa ikon-ikon yang saling terhubung dan memiliki alur

pemrograman. Block diagram dan front panel merupakan dua bagian yang saling terhubung dalam pemrograman menggunakan LabVIEW.

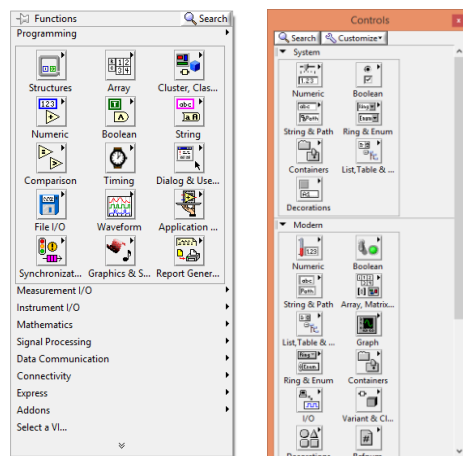


**Gambar 2.6** Tampilan *Front Panel* dan *Block Diagram* pada *LabVIEW*

Sumber : *LabVIEW User Manual*

### 2.2.3. *Control Palette dan Function Palette*

Palet kontrol merupakan kumpulan ikon yang mewakili suatu kontrol yang ada pada front panel, berbagai kontrol dan indikator yang dapat di lihat oleh user seperti kontrol numeris, string, dan lain-lain. Sedangkan *function palette* merupakan kumpulan ikon yang mewakili suatu fungsi pada diagram blok seperti *structure*, *array*, *comparison*, *data communication* dan lain-lain.



**Gambar 2.7** Tampilan *Control Palette* dan *Function Palette*

Sumber : *LabVIEW 2013*

### 2.3 Bahasa Pemrograman *Basic Compiler AVR* (BASCOS AVR)

Bahasa pemrograman basic terkenal didunia sebagai bahasa pemrograman yang handal. Sangat bertolak belakang dari namanya basic, bahasa ini sebenarnya bahasa yang memiliki kemampuan tingkat tinggi. Bahkan banyak para programmer terkenal dunia memakai bahasa pemrograman ini sebagai senjata ampuhnya. Bahasa pemrograman basic banyak digunakan untuk aplikasi mikrokontroler karena kompatibel oleh mikrokontroler jenis AVR dan didukung dengan compiler pemrograman berupa software BASCOS AVR. Bahasa basic memiliki penulisan program yang mudah dimengerti walaupun untuk orang awam sekalipun, karena itu bahasa ini dinamakan bahasa basic. Jenis perintah programnya seperti *do*, *loop*, *if*, *then*, dan sebagainya masih banyak lagi.

BASCOS AVR sendiri adalah salah satu tool untuk pengembangan / pembuatan program untuk kemudian ditanamkan dan dijalankan pada mikrokontroler terutama mikrokontroler keluarga AVR . BASCOS AVR juga bisa disebut sebagai IDE (*Integrated Development Environment*) yaitu lingkungan kerja yang terintegrasi, karena disamping tugas utamanya meng-compile kode program menjadi file hex / bahasa mesin, BASCOS AVR juga memiliki kemampuan / fitur lain yang berguna sekali seperti monitoring komunikasi serial dan untuk menanamkan program yang sudah di compile ke mikrokontroler BASCOS AVR menyediakan pilihan yang dapat mensimulasikan program. Program simulasi ini bertujuan untuk menguji suatu aplikasi yang dibuat dengan pergerakan LED yang ada pada layar simulasi dan dapat juga langsung dilihat pada LCD, jika kita membuat aplikasi yang berhubungan dengan LCD. Intruksi yang dapat digunakan pada editor BASCOS AVR relatif cukup banyak dan tergantung dari tipe dan jenis AVR yang digunakan. Berikut ini adalah beberapa perintah intruksi-intruksi dasar yang digunakan pada BASCOS AVR.

**Tabel 2.1 Intruksi Dasar Bascom AVR**

| Intruksi     | Keterangan         |
|--------------|--------------------|
| DO....LOOP   | Perulangan         |
| GOSUB        | Memanggil prosedur |
| IF....THEN   | Percabangan        |
| FOR.....NEXT | Perulangan         |

|                |                         |
|----------------|-------------------------|
| WAIT           | Waktu tanda detik       |
| WAITMS         | Waktu tanda mili detik  |
| WAITUS         | Waktu tanda micro detik |
| GOTO           | Loncat ke alamat memori |
| SELECT....CASE | Percabangan             |

### 2.3.1 Kontruksi bahasa BASIC pada BASCOM AVR

Setiap bahasa pemrograman mempunyai standar penulisan program. Konstruksi dari program bahasa BASIC harus mengikuti aturan sebagai berikut:

\$regfile = "header"

'inisialisasi

'deklarasi variabel

'deklarasi konstanta

Do

'pernyataan-pernyataan

Loop

end

### 2.3.2 Pengarah Prosesor

\$regfile = "m16def.dat" merupakan pengarah pengarah preprosesor bahasa BASIC yang memerintahkan untuk meyisipkan file lain, dalam hal ini adalah file m16def.dat yang berisi deklarasi register dari mikrokonroller ATmega 16, pengarah preprosesor lainnya yang sering digunakan ialah sebagai berikut:

\$crystal = 12000000 'menggunakan crystal clock 12 MHz

\$baud = 9600 'komunikasi serial dengan baudrate 9600

\$eeprom 'menggunakan fasilitas eeprom

### 2.3.3 Tipe Data

Tipe data merupakan bagian program yang paling penting karena sangat berpengaruh pada program. Pemilihan tipe data yang tepat maka operasi data menjadi lebih efisien dan efektif.

**Tabel 2.2 Tipe Data pada BASCOM AVR**

| No    | Tipe                  | Jangkauan              |
|-------|-----------------------|------------------------|
| 1 – 4 | Bit, Byte dan Integer | 0 atau 10 – 255-32,768 |
| 5     | Word                  | 32,7670 – 65535        |

|          |        |   |
|----------|--------|---|
| <b>6</b> | Long   | -2147483648 – 2147483647                        |
| <b>7</b> | Single | $1.5 \times 10^{-45}$ – $3.4 \times 10^{38}$    |
| <b>8</b> | Double | $5.0 \times 10^{-324}$ to $1.7 \times 10^{308}$ |
|          | String | >254 by   |

#### 2.3.4 Konstanta

Konstanta merupakan suatu nilai dengan tipe data tertentu yang tidak dapat diubah-ubah selama proses program berlangsung. Konstanta harus didefinisikan terlebih dahulu diawal program.

Contoh :         $K_p = 35$ ,  $K_i = 15$ ,  $K_d = 40$

#### 2.3.5 Variabel

Variabel adalah suatu pengenalan (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program yang dapat diubah-ubah sesuai dengan kebutuhan. Nama dari variabel terserah sesuai dengan yang diinginkan namun hal yang terpenting adalah setiap variabel diharuskan :

Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf, max 32 karakter.

Tidak boleh mengandung spasi atau symbol-simbol khusus seperti : \$, ?, %, #, !, &, \*, (, ), -, +, = dan lain sebagainya kecuali underscore.

#### 2.3.6 Deklarasi

Deklarasi sangat diperlukan bila akan menggunakan pengenalan (identifier) dalam suatu program.

##### 2.3.6.1 Deklarasi Variabel

Bentuk umum pendeklarasian suatu variabel adalah Dim nama\_variabel AS tipe\_data

Contoh : Dim x As Integer ‘deklarasi x bertipe integer

##### 2.3.6.2 Deklarasi Konstanta

Dalam Bahasa Basic konstanta di deklarasikan langsung.

Contohnya : S = “Hello world” ‘Assign string

##### 2.3.6.3 Deklarasi Fungsi

Fungsi merupakan bagian yang terpisah dari program dan dapat dipanggil di manapun di dalam program. Fungsi dalam Bahasa Basic ada yang sudah



disediakan sebagai fungsi pustaka seperti print, input data dan untuk menggunakannya tidak perlu dideklarasikan.

#### **2.3.6.4 Deklarasi buatan**

Fungsi yang perlu dideklarasikan terlebih dahulu adalah fungsi yang dibuat oleh programmer. Bentuk umum deklarasi sebuah fungsi adalah :

Sub Test ( byval variabel As type)

Contohnya : Sub Pwm(byval Kiri As Integer , Byval Kanan As Integer)

### **2.3.7 Operator**

#### **2.3.7.1 Operator Penugasan**

Operator Penugasan (Assignment operator) dalam Bahasa Basic berupa “=”.

#### **2.3.7.2 Operator Aritmatika**

\* : untuk perkalian

/ : untuk pembagian

+ : untuk penambahan

- : untuk pengurangan

% : untuk sisa pembagian (modulus)

#### **2.3.7.3 Operator Hubungan (Perbandingan)**

Operator hubungan digunakan untuk membandingkan hubungan dua buah operand atau sebuah nilai / variable, misalnya :

= 'Equality  $X = Y$

< 'Less than  $X < Y$

> 'Greater than  $X > Y$

<= 'Less than or equal to  $X \leq Y$

>= 'Greater than or equal to  $X \geq Y$

#### **2.3.7.4 Operator Logika**

Operator logika digunakan untuk membandingkan logika hasil dari operator-operator hubungan. Operator logika ada empat macam, yaitu :

NOT 'Logical complement

AND 'Conjunction

OR 'Disjunction

XOR 'Exclusive or

### 2.3.7.5 Operator Bitwise

Operator bitwise digunakan untuk memanipulasi bit dari data yang ada di memori. Operator bitwise dalam Bahasa Basic :

Shift A, Left, 2 : Pergeseran bit ke kiri

Shift A, Right, 2 : Pergeseran bit ke kanan

Rotate A, Left, 2 : Putar bit ke kiri

Rotate A, right, 2 : Putar bit ke kanan

## 2.4 Mikrokontroler ATmega 16

Mikrokontroler ATmega 16 merupakan satu jenis mikrokontroler AVR dengan 32 jalur masukan / keluran yang dapat diprogram dalam kemasan 40 pin. Seperti halnya ATmega 162, mikrokontroler ATmega 16 memiliki keistimewaan mikrokontroler jenis AVR, kecuali bahwa memiliki ADC.

### 2.4.1 Beberapa keistimewaan dan fitur – fitur dari mikrokontroler AVR ATmega 16 antara lain :

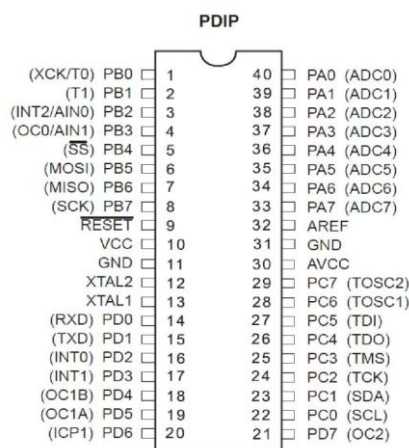
1. *Advanced RISC Architecture*
  - a. *131 powerful Instruction – most single clock cycle execution*
  - b. *32 x 8 general purpose working registers*
  - c. *Fully static operation*
  - d. *Up to 16 MIPS throughput at 16 Mhz*
  - e. *On – chip 2 – cycle multiplier*
2. *High Endurance Non – volatile memory segments*
  - a. *16 Kbytes of In- system self – programmable flash program memory*
  - b. *512 bytes EEPROM*
  - c. *1 Kbyte Internal SRAM*
  - d. *Write / Erase cycles : 10.000 flash / 100.000 EEPROM*
  - e. *Data retention : 20 years at 85° C / 100 years at 25° C (1)*
  - f. *Optional boot code section with independent lock bits in – system programming by on – chip boot program true read – while – write – Operation*
  - g. *Program memory lock for software security*
3. *JTAG ( IEEE std. 1149.1 Compliant ) interface*

- a. *Boudary – scan capabilities according to the JTAG standart*
  - b. *Extensive On- chip debug support*
  - c. *Programming of flash,EEPROM, fuses, and lock bits through the JTAG interface*
4. *Peripheral features*
- a. *Two 8-bit timer / counters with separate prescalers and compare modes*
  - b. *One 16 – bit timer / counter with separate prescaler, compare mode, and capture Mode*
  - a. *Real time counter with separate oscillator*
  - b. *Four PWM Channels*
  - c. *8 – channel ,10 –bit ADC 8 single – ended channels 7 diferential channels in TQFP package only 2 difrential channels with programmbale gain at 1x, 10x or 200x*
  - d. *Byte oriented two – wire serial interface*
  - e. *Programmbale serial USART*
  - f. *Master / Slave SPI Serial Interface*
  - g. *Programmbale watchdog timer with separate on – chip oscillator*
  - h. *On – chip analog comparator*
5. *Special microcontroller features*
- a. *Power – on reset and programmbale brown – out detection*
  - b. *Internal calibrated RC Oscillator*
  - c. *External and Internal Interrupt sources*
  - d. *Six sleep modes : Idle, ADC Noise Reduction, Power –save, power – down , standby and extended standby*
6. *I/O and Packages*
- a. *32 programmbale I/O lines*
  - b. *40 – pin PDIP, 44 – lead TQFP, and 44 – pad QFN / MLF 7*
- Operation Voltages*
7. *Operating Voltages*
- a. *2.7 V – 5.5 V for Atmega 16L*

- b. 4.5 V – 5.5 V for Atmega 16
- 8. Speed Grades
  - a. 0 – 8 MHz for Atmega 16L
  - b. 0 – 16 for Atmega 16
- 9. Power Consumption @ 1 Mhz , 3V , and 25 ° C Atmega 16L
  - a. Active : 1.1 Ma
  - b. Idle mode : 0.35 Ma
  - c. Power-down mode : <1  $\mu$ A

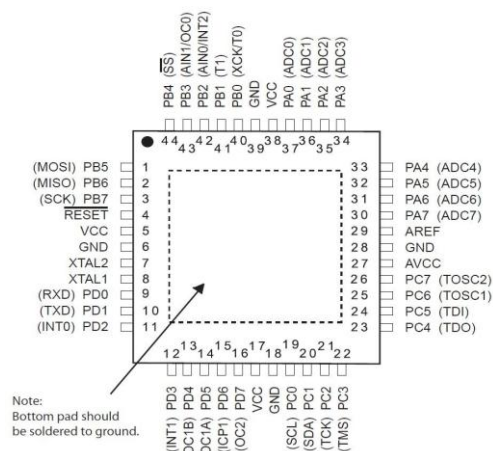
#### 2.4.2 Diagram Pin dan Diagram Blok Mikrokontroler AVR ATmega16

Pada gambar 2.5 dan 2.6 ditunjukkan diagram pin, masing-masing untuk Mikrokontroler AVR ATmega16 tipe PDIP dan TQFP/MLF atau dikenal sebagai SMD.



**Gambar 2.8** Diagram Pin Mikrokontroler AVR ATmega16 tipe PDIP

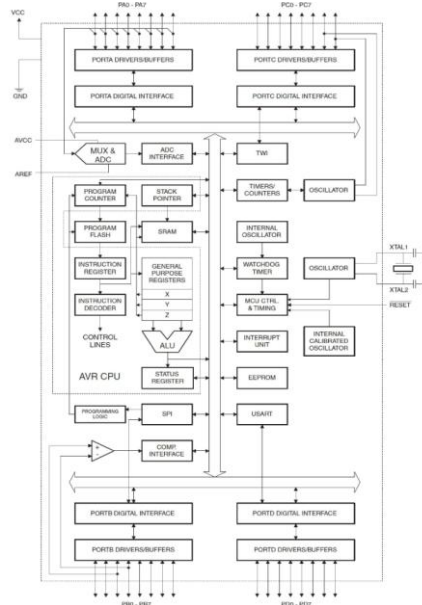
Sumber : <http://www.alldatasheet.com>



**Gambar 2.9** Diagram Pin Mikrokontroler AVR ATmega16 tipe SMD

Sumber : <http://www.alldatasheet.com>

Pada gambar 2.10 ditunjukkan diagram blok untuk Mikrokontroler AVR ATmega16, perhatikan begitu banyaknya fitur-fitur dalam blok tersebut.



**Gambar 2.10** Diagram Blok Mikrokontroler AVR ATmega16

Sumber : <http://www.alldatasheet.com>

### 2.4.3 Fungsi Masing-Masing Pin pada Mikrokontroler AVR ATmega16

Berikut merupakan fungsi dari masing-masing pin pada Mikrokontroler AVR ATmega16 :

- 1 **Vcc** : Masukan tegangan catu daya.
- 2 **GND** : *Ground*.
- 3 **Port A (PA7...PA0)** : Port A berfungsi sebagai masukan analog ke ADC internal

Pada Mikrokontroler AVR ATmega16, selain itu juga berfungsi sebagai port I/O dwi arah 8-bit, jika ADC-nya tidak digunakan. Masing-masing pin menyediakan resistor pull-up internal yang bisa diaktifkan untuk masing-masing bit.

- 4 **Port B (PB7..PB0)** : Port B berfungsi sebagai port I/O dwi-arah 8-bit. Masing-masing pin menyediakan resistor pull-up internal yang bisa diaktifkan

funtuk masing-masing bit.Port B juga memiliki berbagai macam fungsi alternatif, sebagaimana ditunjukkan pada Tabel 1.1.

- |    |                          |   |
|----|--------------------------|---|
| 5  | <b>Port C (PC7..PC0)</b> | : Port C berfungsi sebagai sebagai port I/O dwi-arah 8-bit. Masing-masing pin menyediakan resistor <i>pull-up</i> internal yang bisa diaktifkan untuk masing-masing bit.Port C juga digunakan sebagai antarmuka JTAG, sebagaimana ditunjukkan pada Tabel 1.2. |
| 7  | <b>/RESET</b>            | : Masukan Reset. Level rendah pada pin ini selama lebih dari lama waktu minimum yang ditentukan akan menyebabkan reset, walaupun <i>clock</i> tidak dijalankan.   |
| 8  | <b>XTAL1</b>             | : Masukan ke penguat osilator terbalik ( <i>inverting</i> ) dan masukan ke rangkaian <i>clock</i> internal.   |
| 9  | <b>XTAL2</b>             | : Luaran dari penguat osilator terbalik   |
| 10 | <b>AVCC</b>              | : Merupakan masukan tegangan catu daya untuk Port A sebagai ADC, biasanya dihubungkan ke Vcc, walaupun ADC-nya tidak digunakan. Jika ADC digunakan sebaiknya dihubungkan ke Vcc melalui tapis lolos-bawah ( <i>low-pass filter</i> ).                         |
| 11 | <b>AREF</b>              | : Merupakan tegangan referensi untuk ADC.   |

Pada tabel 2.1, 2.2 dan 2.3 ditunjukkan masing-masing alternatif fungsi dari port B,port C dan port D. Hal ini merupakan penjelasan dari masing-masing fungsi tersebut (kecuali antarmuka JTAG).

**Tabel 2.1** Alternatif Fungsi Port B

| Port Pin | Alternate Functions   |
|----------|---|
| PB7      | SCK (SPI Bus Serial Clock)  |
| PB6      | MISO (SPI Bus Master Input/Slave Output)  |
| PB5      | MOSI (SPI Bus Master Output/Slave Input)  |
| PB4      | SS (SPI Slave Select Input)   |
| PB3      | AIN1 (Analog Comparator Negative Input)<br>OC0 (Timer/Counter0 Output Compare Match Output) |
| PB2      | AIN0 (Analog Comparator Positive Input)<br>INT2 (External Interrupt 2 Input)                |
| PB1      | T1 (Timer/Counter1 External Counter Input)  |
| PB0      | T0 (Timer/Counter0 External Counter Input)<br>XCK (USART External Clock Input/Output)       |

**Tabel 2.2** Alternatif Fungsi Port C

| Port Pin | Alternate Function                               |
|----------|--|
| PC7      | TOSC2 (Timer Oscillator Pin 2)                   |
| PC6      | TOSC1 (Timer Oscillator Pin 1)                   |
| PC5      | TDI (JTAG Test Data In)                          |
| PC4      | TDO (JTAG Test Data Out)                         |
| PC3      | TMS (JTAG Test Mode Select)                      |
| PC2      | TCK (JTAG Test Clock)                            |
| PC1      | SDA (Two-wire Serial Bus Data Input/Output Line) |
| PC0      | SCL (Two-wire Serial Bus Clock Line)             |

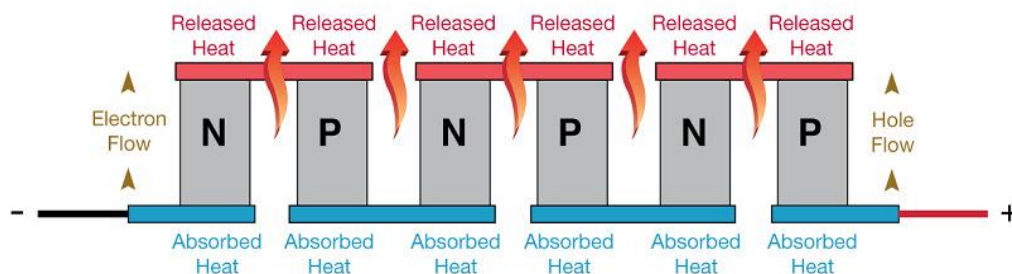
**Tabel 2.3** Alternatif Fungsi Port D

| Port Pin | Alternate Function                                  |
|----------|---|
| PD7      | OC2 (Timer/Counter2 Output Compare Match Output)    |
| PD6      | ICP1 (Timer/Counter1 Input Capture Pin)             |
| PD5      | OC1A (Timer/Counter1 Output Compare A Match Output) |
| PD4      | OC1B (Timer/Counter1 Output Compare B Match Output) |
| PD3      | INT1 (External Interrupt 1 Input)                   |
| PD2      | INT0 (External Interrupt 0 Input)                   |
| PD1      | TXD (USART Output Pin)                              |
| PD0      | RXD (USART Input Pin)                               |

## 2.5 Modul Peltier

Modul peltier adalah komponen elektronik berbasis semikonduktor yang memiliki fungsi sebagai pompa panas. Modul peltier dibuat menggunakan dua tipe semi konduktor yaitu tipe N dan tipe P .Bahan semikonduktor yang dipakai biasanya *Bismuth Telluride*. Modul peltier dirancang dengan menggunakan dua tipe semikonduktor yang dipasang paralel pada termal dan dipasang seri secara elektris.

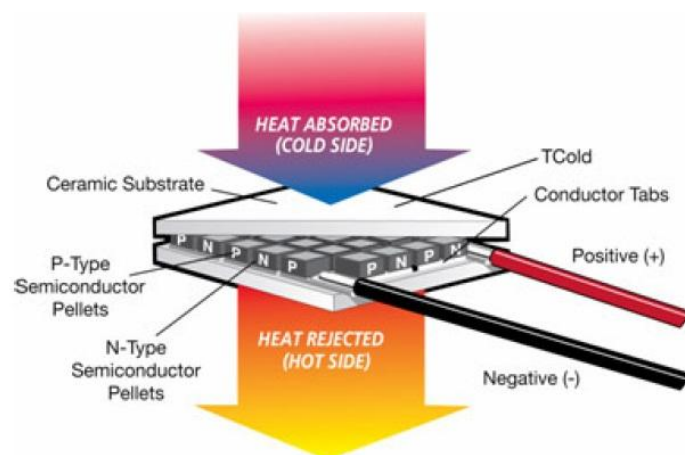
Prinsip kerja modul peltier berdasarkan efek peltier dimana, ketika ada dua buah konduktor berbeda jenis dialiri oleh arus listrik maka akan terbentuk perbedaan temperatur di kedua sisinya. Suhu panas akan dipindahkan menuju arah arus listrik mengalir.



**Gambar 2.11** Prinsip Kerja Peltier

Sumber : <http://www.tellurex.com/technology/peltier-faq.php>

Konstruksi modul peltier juga dilengkapi dengan keramik substrat yang tipis sebagai bahan konduktif untuk mengalirkan suhu yang dihasilkan oleh peltier. Keramik subtrat menjadi standard dalam industri modul peltier sesuai dengan pertimbangan kekuatan mekanis, biaya dan kemampuan konduktifitas.



**Gambar 2.12** Konstruksi Peltier

Sumber : <http://www.tellurex.com/technology/peltier-faq.php>



Peltier memiliki berbagai tipe yang bisa diidentifikasi dari huruf yang tertera pada peltier, seperti pada gambar berikut:

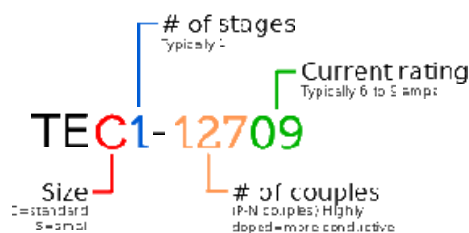
Kelebihan dari peltier antara lain:

1. Tidak mempunyai bagian yang bergerak.
2. Tidak mengandung lapisan ozon.
3. Bentuknya kecil
4. Mudah dikontrol dengan mengubah besarnya tegangan masukan.

Kelemahan dari peltier antara lain:

1. Kemampuan pendinginan atau penyerapan panas yang terbatas bila dibandingkan dengan refrigerasi
2. Memiliki efisiensi yang rendah

Peltier memiliki berbagai tipe yang bisa diidentifikasi dari huruf yang tertera pada peltier, seperti pada gambar berikut:



**Gambar 2.13** Identifikasi Peltier

Sumber : <http://www.tellurex.com/technology/peltier-faq.php>

Kelebihan dari peltier antara lain:

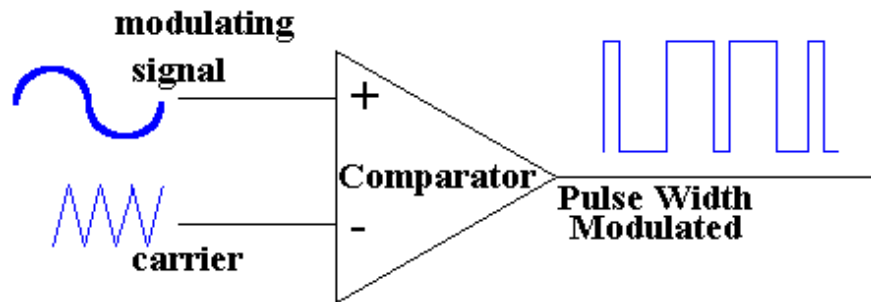
1. Tidak mempunyai bagian yang bergerak.
2. Tidak mengandung lapisan ozon.
3. Bentuknya kecil
4. Mudah dikontrol dengan mengubah besarnya tegangan masukan.

Kelemahan dari peltier antara lain:

1. Kemampuan pendinginan atau penyerapan panas yang terbatas bila dibandingkan dengan refrigerasi
2. Memiliki efisiensi yang rendah

## 2.6 *Pulsa Width Modulation ( PWM )*

PWM atau modulasi lebar pulsa adalah metode modulasi yang mengubah – ubah lebar pulsa pada keluaran berdasarkan sinyal pemodulasinya. Metode yang paling sederhana untuk menghasilkan sinyal PWM berfrekuensi tetap adalah dengan cara menggunakan rangkaian komparator yang memiliki sinyal referensi berbentuk gigi gergaji.



**Gambar 2.14** Komparator PWM

Sumber : <https://insansainsprojects.com>

Ketika amplitudo sinyal pemodulasi lebih kecil dari pada amplitudo carrier maka keluaran komparator menuju level rendah dan sebaliknya ketika amplitudo sinyal pemodulasi lebih besar dari pada amplitudo *carrier* maka keluaran komparator menuju level tinggi.

Terlihat pada gambar, bahwa sinyal PWM adalah sinyal digital yang amplitudonya tetap, namun lebar pulsa yang aktif per periodenya dapat diubah-ubah. Dimana periodenya adalah waktu pulsa *high* (1)  $T_{on}$  ditambah waktu pulsa *low* (0)  $T_{off}$ .

$$T_{total} = T_{on} + T_{off}$$

*Duty cycle* adalah lamanya pulsa *high* (1)  $T_{on}$  dalam satu periode. Jika  $f(t)$  adalah sinyal PWM, maka besar *duty cycle*-nya adalah :

$$\bar{y} = \frac{1}{T} \int_0^T f(t) dt$$

atau bisa ditulis dengan :

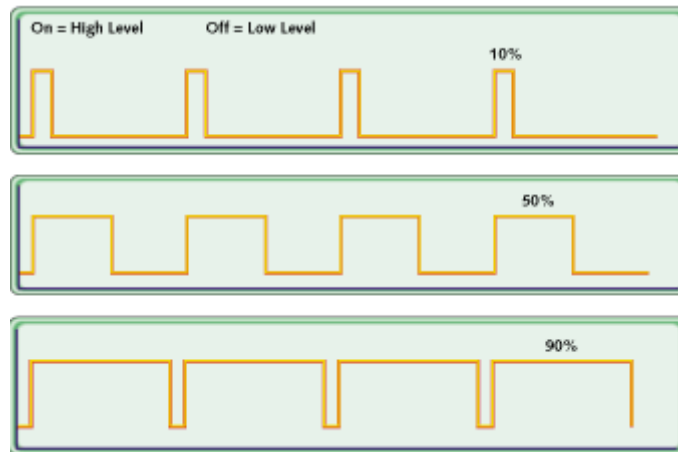
$$D = \frac{T_{on}}{T_{on} + T_{off}} = \frac{T_{on}}{T_{total}}$$

Sehingga

$$V_{out} = D \times V_{in}$$

$$V_{out} = \frac{T_{on}}{T_{total}} \times V_{in}$$

Grafik dibawah ini, menggambarkan beberapa PWM dalam *duty cycle* yang berbeda.



**Gambar 2.15** Grafik PWM

Sumber : <https://insansainsprojects.com>

Pada grafik PWM teratas terlihat bahwa sinyal *high* per periodenya, sangat kecil (hanya 10%). Pada grafik PWM ditengah terlihat sinyal *high*-nya hampir sama dengan sinyal *low* (50%). Dan pada gambar paling bawah terlihat bahwa sinyal *high*-nya lebih besar dari sinyal *low*-nya (90%).

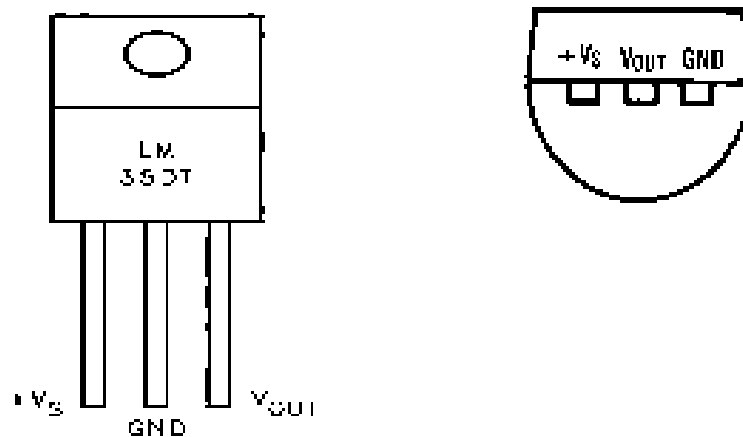
## 2.7 Sensor LM 35

Sensor suhu LM35 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan. Sensor Suhu LM35 yang dipakai dalam penelitian ini berupa komponen elektronika elektronika yang diproduksi oleh *National Semiconductor*. LM35 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, LM35 juga mempunyai keluaran impedansi yang rendah dan linieritas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kendali khusus serta tidak memerlukan penyetelan lanjutan.

Meskipun tegangan sensor ini dapat mencapai 30 volt akan tetapi yang diberikan kesensor adalah sebesar 5 volt, sehingga dapat digunakan dengan catu daya tunggal dengan ketentuan bahwa LM35 hanya membutuhkan arus sebesar 60  $\mu$ A hal ini berarti LM35 mempunyai kemampuan menghasilkan panas (*self-*

*heating*) dari sensor yang dapat menyebabkan kesalahan pembacaan yang rendah yaitu kurang dari 0,5 °C pada suhu 25 °C .

### 2.7.1 Struktur Sensor LM35

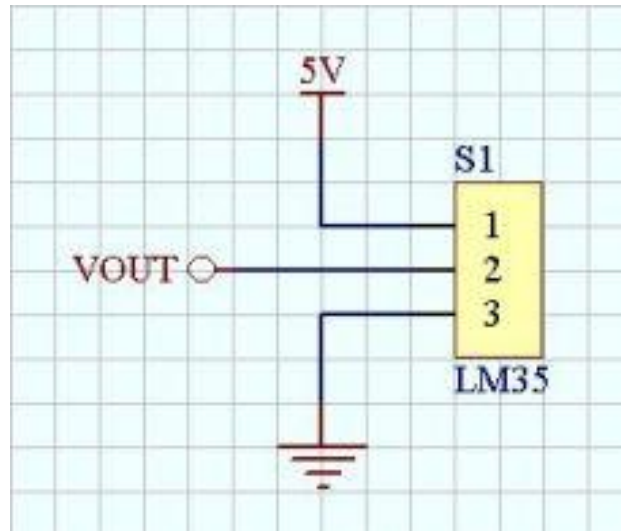


**Gambar 2.16** Sensor Suhu LM35

Sumber : <https://insansainsprojects.com>

Gambar diatas menunjukan bentuk dari LM35 tampak depan dan tampak bawah. 3 pin LM35 menunjukan fungsi masing-masing pin diantaranya, pin 1 berfungsi sebagai sumber tegangan kerja dari LM35, pin 2 atau tengah digunakan sebagai tegangan keluaran atau  $V_{out}$  dengan jangkauan kerja dari 0 Volt sampai dengan 1,5 Volt dengan tegangan operasi sensor LM35 yang dapat digunakan antar 4 Volt sampai 30 Volt. Keluaran sensor ini akan naik sebesar 10 mV setiap derajat *celcius* sehingga diperoleh persamaan sebagai berikut :

$$V_{LM35} = \text{Suhu} * 10 \text{ mV}$$



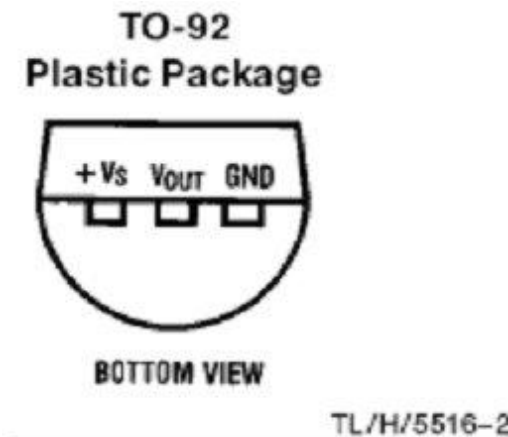
**Gambar 2.17** Skema Sensor LM 35

Sumber : <https://insansainsprojects.com>

Gambar diatas adalah gambar skematik rangkaian dasar sensor suhu LM35-DZ. Rangkaian ini sangat sederhana dan praktis. *Vout* adalah tegangan keluaran sensor yang terskala linear terhadap suhu terukur, yakni 10 milivolt per 1 derajat celcius. Jadi jika *Vout* = 530mV, maka suhu terukur adalah 53 derajat Celcius. Dan jika *Vout* = 320mV, maka suhu terukur adalah 32 derajat Celcius. Tegangan keluaran ini bisa langsung diumpankan sebagai masukan ke rangkaian pengkondisi sinyal seperti rangkaian penguat operasional dan rangkaian filter, atau rangkaian lain seperti rangkaian pembanding tegangan dan rangkaian *Analog-to-Digital Converter*.

Rangkaian dasar tersebut cukup untuk sekedar bereksperimen atau untuk aplikasi yang tidak memerlukan akurasi pengukuran yang sempurna. Akan tetapi tidak untuk aplikasi yang sesungguhnya. Terbukti dari eksperimen yang telah saya lakukan, tegangan keluaran sensor belumlah stabil. Pada kondisi suhu yang relatif sama, jika tegangan suplai saya ubah-ubah (saya naikkan atau turunkan), maka *Vout* juga ikut berubah. Memang secara logika hal ini sepertinya benar, tapi untuk instrumentasi hal ini tidaklah diperkenankan. Dibandingkan dengan tingkat kepresisian, maka tingkat akurasi alat ukur lebih utama karena alat ukur dapat dijadikan patokan bagi penggunaanya. Jika nilainya berubah-ubah untuk kondisi yang relatif tidak ada perubahan, maka alat ukur yang demikian ini tidak dapat digunakan.

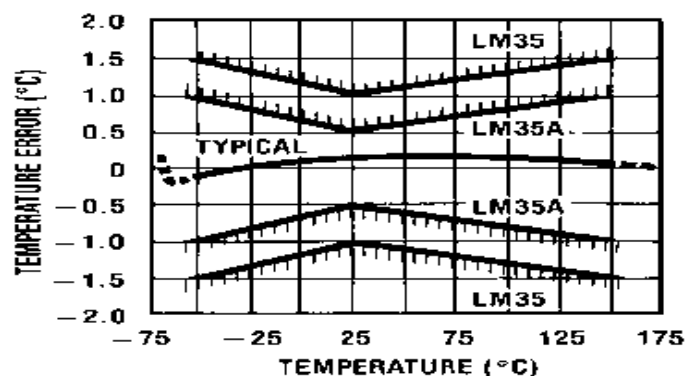
### 2.7.2 Karakteristik Sensor LM35



**Gambar 2.18** Karakteristik Sensor LM 35

Sumber : <https://insansainsprojects.com>

1. Memiliki sensitivitas suhu, dengan faktor skala linier antara tegangan dan suhu 10 mVolt/°C, sehingga dapat dikalibrasi langsung dalam celcius.
2. Memiliki ketepatan atau akurasi kalibrasi yaitu 0,5°C pada suhu 25 °C seperti terlihat pada gambar 2.17
3. Memiliki jangkauan maksimal operasi suhu antara -55 °C sampai +150 °C.
4. Bekerja pada tegangan 4 sampai 30 volt.
5. Memiliki arus rendah yaitu kurang dari 60  $\mu$ A.
6. Memiliki pemanasan sendiri yang rendah (*low-heating*) yaitu kurang dari 0,1 °C pada udara diam.
7. Memiliki impedansi keluaran yang rendah yaitu 0,1 W untuk beban 1 mA.
8. Memiliki ketidaklinieran hanya sekitar  $\pm 1/4$  °C.



**Gambar 2.19** Grafik akurasi LM35 terhadap suhu

Sumber : <https://insansainsprojects.com>

Sensor LM35 bekerja dengan mengubah besaran suhu menjadi besaran tegangan. Tegangan ideal yang keluar dari LM35 mempunyai perbandingan  $100^{\circ}\text{C}$  setara dengan 1 volt. Sensor ini mempunyai pemanasan diri (*self heating*) kurang dari  $0,1^{\circ}\text{C}$ , dapat dioperasikan dengan menggunakan power supply tunggal dan dapat dihubungkan antar muka (*interface*) rangkaian control yang sangat mudah.

IC LM 35 sebagai sensor suhu yang teliti dan terkemas dalam bentuk *Integrated Circuit* (IC), dimana output tegangan keluaran sangat linear terhadap perubahan suhu. Sensor ini berfungsi sebagai pegubah dari besaran fisis suhu ke besaran tegangan yang memiliki koefisien sebesar  $10\text{ mV}/^{\circ}\text{C}$  yang berarti bahwa kenaikan suhu  $1^{\circ}\text{C}$  maka akan terjadi kenaikan tegangan sebesar 10 mV.

Untuk mendeteksi suhu digunakan sebuah sensor suhu LM35 yang dapat dikalibrasikan langsung dalam C (celcius), LM35 ini difungsikan sebagai *basic temperature sensor*.

**Adapun keistimewaan dari IC LM 35 adalah :**

- Kalibrasi dalam satuan derajat celcius.
- Linieritas  $+10\text{ mV}/^{\circ}\text{C}$ .
- Akurasi  $0,5^{\circ}\text{C}$  pada suhu ruang.
- Range  $+2^{\circ}\text{C} - 150^{\circ}\text{C}$ .
- Dioperasikan pada catu daya 4 V – 30 V.
- Arus yang mengalir kurang dari 60 Ma

**Kelebihan dan Kelemahan Sensors LM35**

- Kelebihan:
  - a. Rentang suhu yang jauh, antara  $-55$  sampai  $+150^{\circ}\text{C}$
  - b. *Low self-heating*, sebesar  $0.08^{\circ}\text{C}$
  - c. Beroperasi pada tegangan 4 sampai 30 V
  - d. Rangkaian tidak rumit
  - e. Tidak memerlukan pengkondisian sinyal
- Kekurangan:
  - Membutuhkan sumber tegangan untuk beroperasi